APPENDIX A

The following parameters are utilized by the AMB algorithm:

Input: X - the given design matrix (continuous + categorical) (dimension: m x n, m = # of records, n = # of predictors);
y – the dependent/target variable vector (dimension: m x 1)
Output: s – the solution vector (the model parameter vector, including the "bias" term) (dimension: (n1+1) x 1)

```
Step 0
For each continuous predictor
        If (there is any missing observation value)
                Perform Missing Value Substitution
        End

Step 1
For each continuous predictor
        If (exponentially distributed)
                Log-scale the predictor and flag it
                End
        End
        Detect outliers
End

Step2
// Perform Univariate Analysis for all n predictors
If (size(continuous) > 0)
        For each continuous predictor
                Calculate its Pearson's r value (with the target)
        End
End

If (size(categorical) > 0)
        Bin the continuous target variable
        Calculate its Cramer's V value (on the binned target groups)
End

Sort continuous predictors in Pearson's R value
Sort categorical predictors in Cramer's V value
// Assume n = n_conti + n_cate, n_conti = # of continuous, n_cate = # of categorical
If n_conti > 200
        Retain top 135 + ((n_conti – 200)*0.3) (30% continuous with large R values)
Else if 100 < n_conti <= 200
        Retain top 85 + ((n_conti – 100)*0.5) (50% continuous with large R values)
Else if 50 < n_conti <=100
```

```
        Retain top 50 + ((n_conti - 50)*0.7) (70% continuous with large R values)
Else // n_conti <=50
        Retain all predictors
End
If n_cate > 200
        Retain top 135 + ((n_cate – 200)*0.3) (30% categorical with large V values)
Else if 100 < n_cate <= 200
        Retain top 85 + ((n_cate – 100)*0.5) (50% categorical with large V values)
Else if 50 < n_cate <=100
        Retain top 50 + ((n_cate - 50)*0.7) (70% categorical with large V values)
Else // n_cate <=50
        Retain all predictors
End
```

Step 3

```
If (size(categorical) > 0 & size(continuous)>0)
        // Merge categorical with continuous (in favor of continuous)
        Categorize continuous predictors
                For each categorical predictor c1
                For each continuous predictor c2
                        Compute the Cramer's V value between c1 and c2
                                If Cramer V(c1, c2) > 0.5
                                Remove c1 from the retained list
                        End
                End
        End
End

If (size(categorical) > 0)
        Expand all retained categorical predictors into dummies
End
If (size(categorical) > 0 && size(continuous) > 0)
        Formulate the new design matrix X by combining retained categorical and continuous
        predictors
End
```

Step 4

Normalize (not z-scaling) all retained predictors (X) and obtain the new design matrix X'

Step 5

Formulate the normal equation $N = X'^{T} \cdot X'$ (matrix-matrix multiplication, dimension of N : n1 x n1)

```
//Filter out strongly collinear predictors
While there is an off-diagonal-element of lower_triangle(X'^T·X') with its absolute value > 0.8
        // assume the index is (i, j) and i > j
        Compute the correlation r_i between the target and the ith predictor
```

Compute the correlation r_j between the target and the jth predictor
If r_i > r_j
Remove jth predictor from the retaining predictor list
Else
Remove ith predictor from the retaining predictor list
End
End
If any predictor deletion (above) performed
Reformulate the design matrix X' and the corresponding normal equation $N = X'^{T} \cdot X'$ (matrix-matrix multiplication)
[m, n1] = size(X')
End

Step 6

Perform PCA on N via SVD(N) and obtain the loading matrix M (dimension: n1 x n1) and the latent vector l (dimension: n1 x 1)

Step 7

If PCA successful (i.e., the SVD in PCA does not fail)
Sort the latent vector l in increasing order and obtain the sorting index;
Use singular values l and the sorting index to identify a few bottom components C (i.e., the last d columns of M, dimension: n1 x d) that represents 10 % of variance accounted for;
If (n1 – d < 10)
Reformulate C by including only the last d2 (= n1 – 10) columns of M
Reset d = d2
End
Scan all columns/components in C and delete d1 (<=d) components that don't have a predictive strength, i.e., |Pearson's R(target, component)| <0.3

Step 8

k = n1-d1
Formulate the Mapping matrix M' from M (by removing those d1 components, dimension of M' : n1 x k)
While (k >= m)
Delete the bottom components according to the singular value
End While
Reset k to the size of remaining components
Compute A' = X'M' (matrix-matrix multiplication, dimension of A': m x k)

Step 9

Append the "bias" column (all 1's) to A' as its (new) first column (dimension of A': m x (k+1))
Pass A' to Engine (SVD + possibly a random initial guess and CGD) for component regression and generate a solution vector w (dimension: (k+1) x 1)

Step 10

// Map w back to the predictor space

-- Compute the solution vector s = M' * w [2..k+1] (multiplication of matrix M' and a partial vector of w (from w[2] to w[k+1]) (dimension of s : n1 x 1)

-- Add the "bias" term (i.e., w[1]) to s as its (new) first entry (dimension of s : (n1+1) x 1)

Else // PCA failed

Steps 11

Append the "bias" column to X' as its (new) first column (dimension of X' : m x (n1+1))

While (n+1 >= m)

Delete the remaining least correlated (with target) variable

End While

Reset n+1 to the size of retained design matrix

Pass all retained predictors X' to Engine (SVD + possibly a random initial guess and CGD) for predictor regression and generate a solution vector s (dimension: (n1+1) x 1)

End